



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Edward C. Prosser, et al.

Serial No.: 09/286,862

Filed: April 6, 1999

For: INCORPORATING REGISTER PRESSURE INTO AN
INLINING COMPUTER

Group Art Unit: 2122

Confirmation No.: 5521

Assistant Commissioner for Patents
Washington, DC 20231

APPEAL BRIEF IN SUPPORT OF APPEAL
FROM THE PRIMARY EXAMINER TO THE BOARD OF APPEALS

Sir:

Appellant herewith submits an appeal brief, in triplicate, in support of its appeal to the Board of Appeals from the decision of the Primary Examiner dated October 2, 2002.

The appeal brief fee of \$320.00 is:

☐ Enclosed.

☐ Not required. (Fee paid in prior appeal.)

☒ Charged to Deposit Account No. 09-0465. A duplicate copy of this sheet is enclosed.

Docket No.: RO998-270
Serial No.: 09/286,862



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Edward C. Prosser, et al.

Serial No.: 09/286,862

Filed: April 6, 1999

For: INCORPORATING REGISTER PRESSURE INTO AN
INLINING COMPUTER

Group Art Unit: 2122

Confirmation No.: 5521

Assistant Commissioner for Patents
Washington, DC 20231

APPEAL BRIEF IN SUPPORT OF APPEAL
FROM THE PRIMARY EXAMINER TO THE BOARD OF APPEALS

Sir:

Appellant herewith submits an appeal brief, in triplicate, in support of its appeal to the Board of Appeals from the decision of the Primary Examiner dated October 2, 2002.

The appeal brief fee of \$320.00 is:

- ☐ Enclosed.
- ☐ Not required. (Fee paid in prior appeal.)
- ☒ Charged to Deposit Account No. 09-0465. A duplicate copy of this sheet is enclosed.

Docket No.: RO998-270
Serial No.: 09/286,862

This is an appeal of a Final Rejection of Application Serial No. 09/286,862, filed April 6, 1999. This brief is submitted pursuant to a Notice of Appeal received by the Patent Office on January 7, 2003.

1. Real Party in Interest

The real party in interest, in addition to that indicated in the caption above, is International Business Machines, Inc., the assignee of the above-identified application.

2. Related Appeals and Interferences

There are no related appeals or interferences for the above-identified application.

3. Status of Claims

Claims 1-24 are pending. Appellant appeals from the final rejection of claims 1-4, 9-12, and 17-20 under 35 U.S.C. §103(a) as obvious over U.S. Patent No 5,768,595 to Gillies ("Gillies") in view of U.S. Patent No 5,740,443 to Carini ("Carini"). The Examiner has indicated that claims 5-8, 13-16, and 21-24 would be allowable if rewritten in independent form.

To reduce issues for appeal, Appellant has amended claims 5, 13, and 21 in the Amendment Under MPEP §1207 filed herewith ("§1207 Amendment") to include all of the limitations of the base claims and any intervening claims.

4. Status of Amendments

Appellant amended the claims and drawings in its communication filed April 16, 2002. The Examiner indicated that these amendments were entered and considered in the Office Action dated October 2, 2002.

Appellant has also amended the claims in the §1207 Amendment filed herewith to reduce issues for this Appeal, namely to remove the issues surrounding claims 5-8, 13-16, and 21-24.

5. Summary of Invention

Modern programming techniques commonly utilize many small sequences of instructions, called procedures or objects, each of which is responsible for a particular, well-defined activity. While these techniques can simplify programming effort and reduce complexity, frequent transfers between various procedures and objects (i.e., one procedure or object "calls" another) can degrade program performance because each transfer of control requires multiple computer operations. To address this problem, modern compilers use a technique called inlining. Although the details can be somewhat complex, the idea is simple: the compiler replaces a call to a procedure by a duplicate of the body of the called procedure.

Although inlining provides significant advantages, it also can create problems if overused. One such problem is excessive register pressure. Basically, if the number of registers required at any point (i.e., the pressure) exceeds the number of available physical registers in the processor, some of the values must be maintained in main memory. Because main memory is slower than the registers, such "spillover" significantly degrades system performance. To address this problem, the present invention is directed at a method, system and program product for optimizing compilation by utilizing register-pressure information to make inlining decisions. The present invention overcomes these problems by compiling the source-code twice; once to collect site register-pressure data from the execution; and a second time to make at least one inlining decision using the site register-pressure data.

6. Grouping of Claims

Appellant expressly states that the rejected claims do not stand or fall together. Group I comprises claims 1-4, 9-12, and 17-20. Group II comprises claims 5-8, 13-16, and 21-24.

7. Issue

In view of the §1207 Amendment, Applicant respectfully submits that the sole issue is whether the Examiner has established the required prima facie case of obviousness for the Group I claims (i.e., claims 1-4, 9-12, and 17-20).

8. Argument

Claims 2-4, 10-12, and 18-20 are dependent on claims 1, 9, and 12, respectively, and include all of their limitations. Accordingly, claims 2-4, 10-12, and 18-20 are not made obvious by the cited references if claims 1, 9, and 17 are not made obvious by the cited references.

MPEP §2143.03. Applicant respectfully submits that claims 1, 9, and 17 are not made obvious by the cited references at least because: (i) neither reference discloses making at least one inlining decision using site register-pressure data during a second compilation; and (ii) even if the combination disclosed the claimed inventions, there is no suggestion to make the combination.

(i) Neither reference discloses making at least one inlining decision using site register-pressure data during a second compilation

The PTO has the burden under section 103 to establish a prima facie case of obviousness. *In re Fine*, 5 USPQ2d 1596, 1599 (Fed. Cir. 1988). In order to meet this burden, the Examiner must show that the combination of references suggests each and every element of the claimed invention. *Id.* at 1599; *MPEP* §706.02(j). Thus, there are two fundamental conditions precedent to a proper modification or combination. First, the Examiner must somehow account for all of the elements of the rejected claims. Second the Examiner must be correct when asserting that given elements and limitations in the claims correspond to a particular aspect or aspects of one or more of the cited references. In the case of the Group I claims, the Examiner has failed to establish the first condition precedent because neither reference discloses "making at least one inlining decision using the site register-pressure data during a second compilation."

More specifically, Gillies is directed at an optimizing compiler that controls register usage by identifying procedures that are rarely or never called. *Gillies*, col. 2, lines 50-65. Gillies identifies the target procedures by collecting summary information for each procedure. *Gillies*, col. 4, lines 5-10. This information can include "register pressure/coloring (usage) information." Unlike the present invention, however, Gillies does not collect site register pressure data, much less teach or suggest "making at least one inlining decision using the site

register-pressure data during a second compilation" (emphasis added). Instead, Gillies specifically states that it collects its profile information at the file or procedure level. *Gillies, col. 4, lines 5-11* (stating that "[s]ummary information is collected for the file and each procedure is collected during the compilation. . . ."). Moreover, because the system in Gillies is directed at identifying seldom used child procedures, Gillies would not be interested in collecting site register pressure data. Once the system in Gillies determines that a child procedure is 'seldom used,' it simply ignores that procedure's register pressure during recompilation of any parent procedures. *Gillies, col. 2, lines 35-45*.

Carini also fails to disclose "making at least one inlining decision using the site register-pressure data during a second compilation." Instead, Carini discloses an automatic inlining method that makes decisions based on a cost function. The inputs to the cost function include measurements of the size of the procedure, complexity, number of calls, and number of I/O calls for each procedure; the size of the region containing the call; the loop nesting level of the call; the number of conditionals along the path to the call; and whether or not any of the parameters of the call are used in loop bounds, array bounds, or conditionals of the called procedure. *Carini, col. 7, line 65 - col. 8, line 7*. Like Gillies, however, Carini fails to disclose the use of site register pressure data to make inlining decisions.

(ii) Even if the combination disclosed the claimed invention, there is no suggestion to make the combination.

(a) Carini teaches away from the suggested combination.

The Federal Circuit has held that the fact that a reference that teaches away from the combination is strong evidence of nonobviousness. *In re Fine* at 1599. In this case, Carini is directed at "a long felt need for a call-site specific selective automatic inlining method which does not rely on profiling information." *Carini, col. 5, lines 60-65 (emphasis added); see also Carini, col. 6, lines 10-15* ("It is a further object of the present invention to provide an automatic inlining method which provides performance equivalent to that obtained by user directed inlining . . . but which does not rely on profiling information"). Because Carini specifically teaches away

from using the profile information collected in Gillies, Applicant respectfully submits that there is no motivation to make the suggested combination.

(b) The Examiner is using impermissible hindsight reasoning

The Examiner has also failed to establish the required prima facie case of obviousness because the Examiner improperly used hindsight obtained from Applicant's application to modify Gillies, since there is no basis or suggestion in Gillies to so modify the reference to arrive at the claimed invention. In this case, the Examiner relies on Gillies, col. lines 24-27 as showing a suggestion to make the modification. *Office Action dated October 2, 2002 at §§ 8(i)(c), 8(ii), and 10*. Taken in context, the cited paragraph teaches that summary information for a procedure is collected during compilation, including "[i]nformation as to suitability for other optimization techniques i.e., inlining and cloning, such as intermediate language size, automatic storage usage, and constant parameters, for each procedure." *Gillies, col. 4, lines 5-10 and 24-27*. Contrary to the Examiner's conclusion, however, the cited paragraph would not lead a person skilled in the art to make the suggested modification. Gillies specifically lists the information that a person skilled in the art should collect to determine suitability for other optimization techniques, and this list does not include register pressure information. The fact that Gillies mentions register pressure information only a few sentences after the cited language only reinforces this conclusion. Under the long-established doctrine of *expressio unius est exclusio alterius*, Gillies failure to list register pressure information among the "information as to suitability for other optimization techniques" cannot be considered merely accidental. Accordingly, Appellant respectfully submits that the suggested modification is improper.

To the extent that Examiner argues that the suggested combination obviates the present invention merely because Gillies collects information as to suitability for inlining and that Carini discloses an inlining method, Appellant respectfully submits that the Examiner is improperly using Appellant's own disclosure as a blue print to reconstruct the claimed invention. *Grain Processing Corp. v. American Maize-Products*, 5 USPQ2d 1788, 1792 (Fed. Cir. 1988). Clearly, the combination of Gillies and Carini does not obviate any invention that uses information to


optimize inlining. Instead, the prior art must suggest the desirability of the modification. *In re Fritch*, 23 USPQ2d 1780, 1783-84 (Fed. Cir. 1992). In this case, Appellant respectfully submits that Gillies does not teach or suggest any use for the register pressure information, much less its desirability for use in making inlining decisions. The only suggestion to use site register pressure information to make inlining decisions is found in Appellant's own specification.

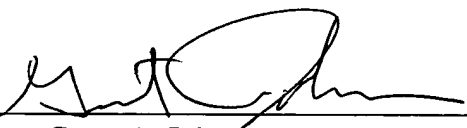
9. Summary

For at least the foregoing reasons, it is submitted that the Examiner's rejections of the Group I claims were erroneous, and reversal of his decisions is respectfully requested.

Date: March 7, 2003

Respectfully submitted,

<p style="text-align: center;">CERTIFICATE OF MAILING UNDER 37 CFR 1.8(a)</p> <p>I hereby certify that the enclosed or attached correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231, on March 7, 2003.</p> <p style="text-align: center;"> _____ Lisa M. Plank</p>

By 
Grant A. Johnson
Registration No.: 42,696

From:
IBM Corporation
Intellectual Property Law
Dept. 917, Bldg. 006-1
3605 Highway 52 North
Rochester, MN 55901

Telephone: (507)253-4660
Fax: (507)253-2382

APPENDIX A (Claims)

1. A method for optimizing a computer program comprising a child procedure and a parent procedure, wherein the parent procedure comprises at least one statement that invokes the child procedure, wherein the method comprises:

 saving site register-pressure data from the execution of a first compilation; and
 making at least one inlining decision using the site register-pressure data during a second compilation.

2. The method of claim 1, further comprising:

 saving a maximum register-pressure occurring in each procedure in the computer program; and
 making the at least one inlining decision using the maximum register-pressure data.

3. The method of claim 2, wherein the site register-pressure data comprises:

 a register pressure at each call site in the computer program that is a potential inlining candidate.

4. The method of claim 1, wherein the making step further comprises:

 inlining the child procedure of the computer program into the parent procedure, in place of the statement that invokes the child procedure.

5. A method for optimizing a computer program comprising a child procedure and a parent procedure, wherein the parent procedure comprises at least one statement that invokes the child procedure, wherein the method comprises:

saving site register-pressure data from the execution of a first compilation, wherein the site register pressure data comprises a register pressure at each call site in the computer program that is a potential inlining candidate; and

making at least one inlining decision using the site register-pressure data during a second compilation, wherein when deciding whether to inline the child procedure into the parent procedure, determining whether a sum of the maximum register-pressure and the site register-pressure exceeds a number of available registers.

6. The method of claim 5, further comprising:

when the determining step is true, refraining from inlining the child procedure into the parent procedure.

7. The method of claim 5, further comprising:

when the determining step is false, inlining the child procedure into the parent procedure in place of the statement that invokes the child procedure.

8. The method of claim 7, further comprising:

setting the maximum register-pressure of the parent procedure to be a maximum of its existing value or the sum of the maximum register-pressure of the child procedure and the site register-pressure.

9. A computer system for compiling a computer program including a child procedure and a parent procedure which includes one or more statements that invoke the child procedure, into a machine-readable representation, the computer system comprising:

an optimizer that optimizes the computer program into an optimized representation, the optimizer saving site register-pressure data from the execution of a first compilation, and making at least one inlining decision using the site register-pressure data during a second compilation; and

a machine-readable code generator that generates a machine-readable representation of the computer procedure from the optimized representation.

10. The computer system of claim 9, wherein the optimizer further:
saves a maximum register-pressure occurring in each procedure in the computer program;
and
makes the at least one inlining decision using the maximum register pressure data.

11. The computer system of claim 10, wherein the site register-pressure data comprises:
a register pressure at each call site in the computer program that is a potential inlining candidate.

12. The computer system of claim 9, wherein the optimizer further comprises:
inlining the child procedure of the computer program into the parent procedure, in place of the statement that invokes the child procedure.

13. A computer system for compiling a computer program including a child procedure and a parent procedure which includes one or more statements that invoke the child procedure, into a machine-readable representation, the computer system comprising:

an optimizer that optimizes the computer program into an optimized representation, the optimizer saving site register-pressure data from the execution of a first compilation, and making at least one inlining decision using the site register-pressure data during a second compilation;

wherein the site register-pressure data comprises a register pressure at each call site in the computer program that is a potential inlining candidate; and

wherein, when deciding whether to inline the child procedure into the parent procedure, the optimizer further determines whether a sum of the maximum register-pressure and the site register-pressure exceeds a number of available registers; and

a machine-readable code generator that generates a machine-readable representation of the computer procedure from the optimized representation.

14. The computer system of claim 13, wherein the optimizer further comprises:
when the determining step is true, refraining from inlining the child procedure into the parent procedure.

15. The computer system of claim 13, wherein the optimizer further comprises:
when the determining step is false, inlining the child procedure into the parent procedure in place of the statement that invokes the child procedure.

16. The computer system of claim 15, wherein the optimizer further comprises:
setting the maximum register-pressure of the parent procedure to be a maximum of its existing value or the sum of the maximum register-pressure of the child procedure and the site register-pressure.

17. A program product for optimizing a computer program that includes a child procedure and a parent procedure that comprises at least one statement that invokes the child procedure, comprising:

an optimizer that saves site register-pressure data from the execution of a first compilation, and

makes at least one inlining decision using the site register-pressure data during a second compilation; and

signal-bearing media bearing the optimizer.

18. The program product of claim 17, wherein the optimizer further:
saves a maximum register-pressure occurring in each procedure in the computer program;
and

makes the at least one inlining decision using the maximum register-pressure data.

19. The program product of claim 18, wherein the site register-pressure data comprises:
a register pressure at each call site in the computer program that is a potential inlining candidate.

20. The program product of claim 17, wherein the optimizer further:
inlines the child procedure of the computer program into the parent procedure, in place of the statement that invokes the child procedure.

21. A program product for optimizing a computer program that includes a child procedure and a parent procedure that comprises at least one statement that invokes the child procedure, comprising:

an optimizer that saves site register-pressure data from the execution of a first compilation, and makes at least one inlining decision using the site register-pressure data during a second compilation;

wherein the site register-pressure data comprises a register pressure at each call site in the computer program that is a potential inlining candidate; and

wherein, when deciding whether to inline the child procedure into the parent procedure, the optimizer determines whether a sum of the maximum register-pressure and the site register-pressure exceeds a number of available registers; and

signal-bearing media bearing the optimizer.

22. The program product of claim 21, wherein the optimizer further:

when the determining step is true, refrains from inlining the child procedure into the parent procedure.

23. The program product of claim 21, wherein the optimizer further:

when the determining step is false, inlines the child procedure into the parent procedure in place of the statement that invokes the child procedure.

24. The program product of claim 23, wherein the optimizer further:

sets the maximum register-pressure of the parent procedure to be a maximum of its existing value or the sum of the maximum register-pressure of the child procedure and the site register-pressure.

This is an appeal of a Final Rejection of Application Serial No. 09/286,862, filed April 6, 1999. This brief is submitted pursuant to a Notice of Appeal received by the Patent Office on January 7, 2003.

1. Real Party in Interest

The real party in interest, in addition to that indicated in the caption above, is International Business Machines, Inc., the assignee of the above-identified application.

2. Related Appeals and Interferences

There are no related appeals or interferences for the above-identified application.

3. Status of Claims

Claims 1-24 are pending. Appellant appeals from the final rejection of claims 1-4, 9-12, and 17-20 under 35 U.S.C. §103(a) as obvious over U.S. Patent No 5,768,595 to Gillies ("Gillies") in view of U.S. Patent No 5,740,443 to Carini ("Carini"). The Examiner has indicated that claims 5-8, 13-16, and 21-24 would be allowable if rewritten in independent form.

To reduce issues for appeal, Appellant has amended claims 5, 13, and 21 in the Amendment Under MPEP §1207 filed herewith ("§1207 Amendment") to include all of the limitations of the base claims and any intervening claims.

4. Status of Amendments

Appellant amended the claims and drawings in its communication filed April 16, 2002. The Examiner indicated that these amendments were entered and considered in the Office Action dated October 2, 2002.

Appellant has also amended the claims in the §1207 Amendment filed herewith to reduce issues for this Appeal, namely to remove the issues surrounding claims 5-8, 13-16, and 21-24.

5. Summary of Invention

Modern programming techniques commonly utilize many small sequences of instructions, called procedures or objects, each of which is responsible for a particular, well-defined activity. While these techniques can simplify programming effort and reduce complexity, frequent transfers between various procedures and objects (i.e., one procedure or object "calls" another) can degrade program performance because each transfer of control requires multiple computer operations. To address this problem, modern compilers use a technique called inlining. Although the details can be somewhat complex, the idea is simple: the compiler replaces a call to a procedure by a duplicate of the body of the called procedure.

Although inlining provides significant advantages, it also can create problems if overused. One such problem is excessive register pressure. Basically, if the number of registers required at any point (i.e., the pressure) exceeds the number of available physical registers in the processor, some of the values must be maintained in main memory. Because main memory is slower than the registers, such "spillover" significantly degrades system performance. To address this problem, the present invention is directed at a method, system and program product for optimizing compilation by utilizing register-pressure information to make inlining decisions. The present invention overcomes these problems by compiling the source-code twice; once to collect site register-pressure data from the execution; and a second time to make at least one inlining decision using the site register-pressure data.

6. Grouping of Claims

Appellant expressly states that the rejected claims do not stand or fall together. Group I comprises claims 1-4, 9-12, and 17-20. Group II comprises claims 5-8, 13-16, and 21-24.

7. Issue

In view of the §1207 Amendment, Applicant respectfully submits that the sole issue is whether the Examiner has established the required prima facie case of obviousness for the Group I claims (i.e., claims 1-4, 9-12, and 17-20).

8. Argument

Claims 2-4, 10-12, and 18-20 are dependent on claims 1, 9, and 12, respectively, and include all of their limitations. Accordingly, claims 2-4, 10-12, and 18-20 are not made obvious by the cited references if claims 1, 9, and 17 are not made obvious by the cited references.

MPEP §2143.03. Applicant respectfully submits that claims 1, 9, and 17 are not made obvious by the cited references at least because: (i) neither reference discloses making at least one inlining decision using site register-pressure data during a second compilation; and (ii) even if the combination disclosed the claimed inventions, there is no suggestion to make the combination.

(i) Neither reference discloses making at least one inlining decision using site register-pressure data during a second compilation

The PTO has the burden under section 103 to establish a prima facie case of obviousness. *In re Fine*, 5 USPQ2d 1596, 1599 (Fed. Cir. 1988). In order to meet this burden, the Examiner must show that the combination of references suggests each and every element of the claimed invention. *Id.* at 1599; *MPEP* §706.02(j). Thus, there are two fundamental conditions precedent to a proper modification or combination. First, the Examiner must somehow account for all of the elements of the rejected claims. Second the Examiner must be correct when asserting that given elements and limitations in the claims correspond to a particular aspect or aspects of one or more of the cited references. In the case of the Group I claims, the Examiner has failed to establish the first condition precedent because neither reference discloses "making at least one inlining decision using the site register-pressure data during a second compilation."

More specifically, Gillies is directed at an optimizing compiler that controls register usage by identifying procedures that are rarely or never called. *Gillies*, col. 2, lines 50-65. Gillies identifies the target procedures by collecting summary information for each procedure. *Gillies*, col. 4, lines 5-10. This information can include "register pressure/coloring (usage) information." Unlike the present invention, however, Gillies does not collect site register pressure data, much less teach or suggest "making at least one inlining decision using the site

register-pressure data during a second compilation" (emphasis added). Instead, Gillies specifically states that it collects its profile information at the file or procedure level. *Gillies*, col. 4, lines 5-11 (stating that "[s]ummary information is collected for the file and each procedure is collected during the compilation. . . ."). Moreover, because the system in Gillies is directed at identifying seldom used child procedures, Gillies would not be interested in collecting site register pressure data. Once the system in Gillies determines that a child procedure is 'seldom used,' it simply ignores that procedure's register pressure during recompilation of any parent procedures. *Gillies*, col. 2, lines 35-45.

Carini also fails to disclose "making at least one inlining decision using the site register-pressure data during a second compilation." Instead, Carini discloses an automatic inlining method that makes decisions based on a cost function. The inputs to the cost function include measurements of the size of the procedure, complexity, number of calls, and number of I/O calls for each procedure; the size of the region containing the call; the loop nesting level of the call; the number of conditionals along the path to the call; and whether or not any of the parameters of the call are used in loop bounds, array bounds, or conditionals of the called procedure. *Carini*, col. 7, line 65 - col. 8, line 7. Like Gillies, however, Carini fails to disclose the use of site register pressure data to make inlining decisions.

(ii) Even if the combination disclosed the claimed invention, there is no suggestion to make the combination.

(a) Carini teaches away from the suggested combination.

The Federal Circuit has held that the fact that a reference teaches away from the combination is strong evidence of nonobviousness. *In re Fine* at 1599. In this case, Carini is directed at "a long felt need for a call-site specific selective automatic inlining method which does not rely on profiling information." *Carini*, col. 5, lines 60-65 (*emphasis added*); *see also Carini*, col. 6, lines 10-15 ("It is a further object of the present invention to provide an automatic inlining method which provides performance equivalent to that obtained by user directed inlining . . . but which does not rely on profiling information"). Because Carini specifically teaches away

from using the profile information collected in Gillies, Applicant respectfully submits that there is no motivation to make the suggested combination.

(b) The Examiner is using impermissible hindsight reasoning

The Examiner has also failed to establish the required *prima facie* case of obviousness because the Examiner improperly used hindsight obtained from Applicant's application to modify Gillies, since there is no basis or suggestion in Gillies to so modify the reference to arrive at the claimed invention. In this case, the Examiner relies on Gillies, col. lines 24-27 as showing a suggestion to make the modification. *Office Action dated October 2, 2002 at ¶¶ 8(i)(c), 8(ii), and 10*. Taken in context, the cited paragraph teaches that summary information for a procedure is collected during compilation, including "[i]nformation as to suitability for other optimization techniques i.e., inlining and cloning, such as intermediate language size, automatic storage usage, and constant parameters, for each procedure." *Gillies, col. 4, lines 5-10 and 24-27*. Contrary to the Examiner's conclusion, however, the cited paragraph would not lead a person skilled in the art to make the suggested modification. Gillies specifically lists the information that a person skilled in the art should collect to determine suitability for other optimization techniques, and this list does not include register pressure information. The fact that Gillies mentions register pressure information only a few sentences after the cited language only reinforces this conclusion. Under the long-established doctrine of *expressio unius est exclusio alterius*, Gillies failure to list register pressure information among the "information as to suitability for other optimization techniques" cannot be considered merely accidental. Accordingly, Appellant respectfully submits that the suggested modification is improper.

To the extent that Examiner argues that the suggested combination obviates the present invention merely because Gillies collects information as to suitability for inlining and that Carini discloses an inlining method, Appellant respectfully submits that the Examiner is improperly using Appellant's own disclosure as a blue print to reconstruct the claimed invention. *Grain Processing Corp. v. American Maize-Products*, 5 USPQ2d 1788, 1792 (Fed. Cir. 1988). Clearly, the combination of Gillies and Carini does not obviate any invention that uses information to


optimize inlining. Instead, the prior art must suggest the desirability of the modification. *In re Fritch*, 23 USPQ2d 1780, 1783-84 (Fed. Cir. 1992). In this case, Appellant respectfully submits that Gillies does not teach or suggest any use for the register pressure information, much less its desirability for use in making inlining decisions. The only suggestion to use site register pressure information to make inlining decisions is found in Appellant's own specification.

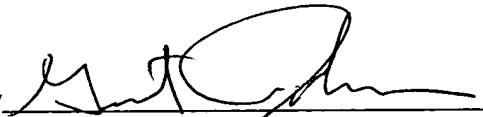
9. Summary

For at least the foregoing reasons, it is submitted that the Examiner's rejections of the Group I claims were erroneous, and reversal of his decisions is respectfully requested.

Date: March 7, 2003

Respectfully submitted,

<p style="text-align: center;">CERTIFICATE OF MAILING UNDER 37 CFR 1.8(a)</p> <p>I hereby certify that the enclosed or attached correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231, on March 7, 2003.</p> <p style="text-align: center;"> Lisa M. Plank</p>

By 
Grant A. Johnson
Registration No.: 42,696

From: IBM Corporation
Intellectual Property Law
Dept. 917, Bldg. 006-1
3605 Highway 52 North
Rochester, MN 55901

Telephone: (507)253-4660
Fax: (507)253-2382

APPENDIX A (Claims)

1. A method for optimizing a computer program comprising a child procedure and a parent procedure, wherein the parent procedure comprises at least one statement that invokes the child procedure, wherein the method comprises:

 saving site register-pressure data from the execution of a first compilation; and
 making at least one inlining decision using the site register-pressure data during a second compilation.

2. The method of claim 1, further comprising:

 saving a maximum register-pressure occurring in each procedure in the computer program; and
 making the at least one inlining decision using the maximum register-pressure data.

3. The method of claim 2, wherein the site register-pressure data comprises:

 a register pressure at each call site in the computer program that is a potential inlining candidate.

4. The method of claim 1, wherein the making step further comprises:

 inlining the child procedure of the computer program into the parent procedure, in place of the statement that invokes the child procedure.

5. A method for optimizing a computer program comprising a child procedure and a parent procedure, wherein the parent procedure comprises at least one statement that invokes the child procedure, wherein the method comprises:

saving site register-pressure data from the execution of a first compilation, wherein the site register pressure data comprises a register pressure at each call site in the computer program that is a potential inlining candidate; and

making at least one inlining decision using the site register-pressure data during a second compilation, wherein when deciding whether to inline the child procedure into the parent procedure, determining whether a sum of the maximum register-pressure and the site register-pressure exceeds a number of available registers.

6. The method of claim 5, further comprising:

when the determining step is true, refraining from inlining the child procedure into the parent procedure.

7. The method of claim 5, further comprising:

when the determining step is false, inlining the child procedure into the parent procedure in place of the statement that invokes the child procedure.

8. The method of claim 7, further comprising:

setting the maximum register-pressure of the parent procedure to be a maximum of its existing value or the sum of the maximum register-pressure of the child procedure and the site register-pressure.

9. A computer system for compiling a computer program including a child procedure and a parent procedure which includes one or more statements that invoke the child procedure, into a machine-readable representation, the computer system comprising:

an optimizer that optimizes the computer program into an optimized representation, the optimizer saving site register-pressure data from the execution of a first compilation, and making at least one inlining decision using the site register-pressure data during a second compilation; and

a machine-readable code generator that generates a machine-readable representation of the computer procedure from the optimized representation.

10. The computer system of claim 9, wherein the optimizer further:

saves a maximum register-pressure occurring in each procedure in the computer program;

and

makes the at least one inlining decision using the maximum register pressure data.

11. The computer system of claim 10, wherein the site register-pressure data comprises:

a register pressure at each call site in the computer program that is a potential inlining candidate.

12. The computer system of claim 9, wherein the optimizer further comprises:

inlining the child procedure of the computer program into the parent procedure, in place of the statement that invokes the child procedure.

13. A computer system for compiling a computer program including a child procedure and a parent procedure which includes one or more statements that invoke the child procedure, into a machine-readable representation, the computer system comprising:

an optimizer that optimizes the computer program into an optimized representation, the optimizer saving site register-pressure data from the execution of a first compilation, and making at least one inlining decision using the site register-pressure data during a second compilation;

wherein the site register-pressure data comprises a register pressure at each call site in the computer program that is a potential inlining candidate; and

wherein, when deciding whether to inline the child procedure into the parent procedure, the optimizer further determines whether a sum of the maximum register-pressure and the site register-pressure exceeds a number of available registers; and

a machine-readable code generator that generates a machine-readable representation of the computer procedure from the optimized representation.

14. The computer system of claim 13, wherein the optimizer further comprises:
when the determining step is true, refraining from inlining the child procedure into the parent procedure.

15. The computer system of claim 13, wherein the optimizer further comprises:
when the determining step is false, inlining the child procedure into the parent procedure in place of the statement that invokes the child procedure.

16. The computer system of claim 15, wherein the optimizer further comprises:
setting the maximum register-pressure of the parent procedure to be a maximum of its existing value or the sum of the maximum register-pressure of the child procedure and the site register-pressure.

17. A program product for optimizing a computer program that includes a child procedure and a parent procedure that comprises at least one statement that invokes the child procedure, comprising:

an optimizer that saves site register-pressure data from the execution of a first compilation, and

makes at least one inlining decision using the site register-pressure data during a second compilation; and

signal-bearing media bearing the optimizer.

18. The program product of claim 17, wherein the optimizer further:

saves a maximum register-pressure occurring in each procedure in the computer program;
and

makes the at least one inlining decision using the maximum register-pressure data.

19. The program product of claim 18, wherein the site register-pressure data comprises:

a register pressure at each call site in the computer program that is a potential inlining candidate.

20. The program product of claim 17, wherein the optimizer further:

inlines the child procedure of the computer program into the parent procedure, in place of the statement that invokes the child procedure.

21. A program product for optimizing a computer program that includes a child procedure and a parent procedure that comprises at least one statement that invokes the child procedure, comprising:

an optimizer that saves site register-pressure data from the execution of a first compilation, and makes at least one inlining decision using the site register-pressure data during a second compilation;

wherein the site register-pressure data comprises a register pressure at each call site in the computer program that is a potential inlining candidate; and

wherein, when deciding whether to inline the child procedure into the parent procedure, the optimizer determines whether a sum of the maximum register-pressure and the site register-pressure exceeds a number of available registers; and

signal-bearing media bearing the optimizer.

22. The program product of claim 21, wherein the optimizer further:

when the determining step is true, refrains from inlining the child procedure into the parent procedure.

23. The program product of claim 21, wherein the optimizer further:

when the determining step is false, inlines the child procedure into the parent procedure in place of the statement that invokes the child procedure.

24. The program product of claim 23, wherein the optimizer further:

sets the maximum register-pressure of the parent procedure to be a maximum of its existing value or the sum of the maximum register-pressure of the child procedure and the site register-pressure.